

PROGRAMMING LANGUAGES: History and Fundamentals

by Jean E. Sammet

**A Unique, Definitive Sourcebook
Presenting You With
120 Programming Languages**

Out of a profusion of languages comes a clear, logical, understandable presentation enabling you to decide which languages you should investigate in more detail, for your particular purpose. And also furnishing you with history and perspective showing you why there are so many languages in use today. Virtually all languages described have been implemented, so they are not just theoretical ideas.

But why is this book so outstanding?

IT'S PROLIFIC, including and discussing more than 120 higher level languages—all of the major and most of the minor ones developed in the United States. Describes their history, general characteristics, similarities and differences. The picturesque Tower of Babel names all the languages described in the book.

And virtually every language description was sent to experts in that language for correction and comment, thus trying to assure you of accuracy and clarity in each presentation.

IT'S UNIQUE, bringing together in one volume, and in consistent fashion, fundamental information on programming languages. Information making it possible for you to understand and write very simple programs in many of the languages.

There is a division of the fundamental aspects of programming languages into technical and non-technical characteristics. These are defined for you, and the concepts are then used consistently for the descriptions of the major languages.

IT'S REFERENCE ORIENTED, containing an extensive (over 800 items) bibliography grouped by language. It includes citations of the major source documents, general descriptions, uses, and related materials. There are 2 indexes and an author list to facilitate your use of the book as a reference source.

(page 4, please)



Concept of the tower of BABEL to represent a large set of programming languages is due to the Communications of the ACM, a publication of the Association for Computing Machinery, Inc.



PROGRAMMING LANGUAGES: History and Fundamentals

Here at last, in a definitive single source, the reader can find basic information about all of the major and most of the minor higher level languages developed in the U.S.

This is just not another collection of previously published articles; rather, it is an original reference work that provides fundamental information on programming languages, including history, general characteristics, similarities, and differences. The fundamental aspects of programming languages are divided into technical and nontechnical characteristics. These are defined and the concepts are then used consistently for the descriptions of the major languages.

The broad coverage, combined with numerous examples and a large bibliography, enables the reader to decide which languages he should investigate in more detail for his particular purpose.

Outstanding Features:

- ☐ Broad coverage—about 120 languages.
- ☐ Over 800 individual bibliographic items.
- ☐ Sample programs show the basic elements of about 30 languages at a glance.
- ☐ Extensive bibliographies for each language, including citations of the major source documents, general description, uses, and related material.
- ☐ Detailed description of the historical development of the major languages.
- ☐ An appendix that includes a list arranged by author showing the page number where the full bibliographic citation can be found, and pages on which it is referenced.
- ☐ Provides history and perspective to show why there are so many languages in use today.
- ☐ An appendix containing a list of each language with the meaning of its acronym, a brief description, relevant subsection number and best references.
- ☐ A very detailed table of contents shows the structure of the programming language field at a glance.

Do You Know What Languages These Are?

MAXIMUM n=20.

READ n.

READ A_{1j} FROM j=1 TO n AND i=1 TO n.

READ C_1 FROM i=1 TO n.

FROM j=1 TO n AND i=1 TO n IF $i > j$ THEN $a_{1j} = A_{1j} - \sum_{k=1}^{j-1} a_{1k} a_{kj}$ OTHERWISE $a_{1j} = \frac{A_{1j} - \sum_{k=1}^{i-1} a_{1k} a_{kj}}{a_{11}}$.

FROM i=1 TO n COMPUTE $\gamma_1 = \frac{C_1 - \sum_{k=1}^{i-1} a_{1k} \gamma_k}{a_{11}}$.

FROM i=n BY -1 UNTIL $i < 1$ COMPUTE $X_1 = \gamma_1 - \sum_{k=i+1}^n a_{1k} X_k$.

PRINT 1 { 2 }, X_1 FOR i=1, 2, ..., n. FINISH.

```
#1,1 /* start automatic line numbering with 1 and */
2. /* with increments of 1 */
3. /* This is a program to calculate mean and */
4. /* standard deviation of a table of numbers */
5. DECLARE p(10),psqr(10)
6. LET s(x)=sqrt(x/n-ii**2) /* define standard deviation*/
7. GET LIST(n,p)
8. psqr=p**2 /* square the entire array */
9. sum# = 0
10. sum2 = 0
11. label: DO i=1 TO n
12. sum# = sum# + p(i)
13. sum2 = sum2 + psqr(i)
14. END label
15. PUT IMAGE(sum#/n,s(sum2)) (im1)
16. im1: IMAGE
Mean=..... Standard Deviation=-----
17. /* .....indicates scientific notation*/
18. /*-----indicates decimal conversion*/
19. GO TO start
20.
p(10)=7/3 /* Set value requiring arithmetic operation*/
execute 1 thru ... /* execute entire program */
n
10
p(1)
2.5,3.5,8.1,10,11,8.2,5.5,7,8.2,,
15. 6911 VALUE OF M IS NOT DEFINED
***** XEQ ERROR.
14.5 M=sum#/n /* Insert correction after statement 14*/
execute 9 thru ... /* Run program from statement 9 to end */
Mean=.7133333E01 Standard Deviation= 2.5581243128
***** 19. "GOTO" OPERAND NOT LABEL.
7 start: GET LIST (n,p)
/* Correct statement number 7 */
p=p-M /* As a test, calculate distance to mean.*/
xeq 9 thru ... /* and do a rerun */
Mean=.444089E-15 Standard Deviation= 7.5781557410
***** 19. "GOTO" TARGET OUTSIDE XEQ RANGE.
```


TABLE OF CONTENTS

I. General Introduction.

Machine Language Programming. Symbolic Assembly Language Programming. Early Development of Better Tools. Specific Needs to be Met. Brief History of Early Efforts. Definition of Programming Languages. Definition Problem. Defining Characteristics. Basic Technology. Difference between Programming Language and Application Package. Advantages and Disadvantages of Higher Level Languages. Advantages. Disadvantages. Overall Evaluation. Classifications of Programming Languages and Proposed Definitions. Procedure-Oriented Language. Non-Procedural Language. Problem-Oriented Language. Application-Oriented Language. Special Purpose Language. Problem-Defining Language. Problem-Describing Language. Problem-Solving Language. Reference Language. Publication Language. Hardware Language. Factors in Choice of a Language. Suitability of Language for Problem Area and Projected Users. Availability on Desired Computer. History and Evaluation of Previous Use. Efficiency of Language Implementation. Compatibility and Growth Potential. Functions. (=Non-Technical Characteristics). Technical Characteristics.

II. FUNCTIONAL CHARACTERISTICS OF PROGRAMMING LANGUAGES.

Description of Concept of Functional Characteristics. Properties of Languages. Purpose of Language. Application Area. Type of Language. Type of User. Physical Environment. Conversion and Compatibility. Types of Compatibility. Ease of Conversion. Standardization. Purposes. Problems. Method of Establishing Standards. Overall Status. Types and Methods of Language Definition. Administrative. Technical. Types of Documentation. Evaluation Based on Use. Availability on Differing Computers. Evaluation of Language vs. Evaluation of Compiler. Usage Relative to Objectives. Advantages. Disadvantages. Mistakes to be Avoided in the Future.

III. TECHNICAL CHARACTERISTICS OF PROGRAMMING LANGUAGES.

Description of Concept of Technical Features. Introduction. Major Parts of a Language. Form of Language. Character Set. Types of Basic Elements (=Tokens). Identifier Definition. Definition and Usage of Other Basic Elements. Type of Input Form Used. Structure of Program. Types of Subunits. Characteristics of Subunits. Data Types and Units and Computations with Them. Types of Data Variables and Constants. Accessible Data Units. Types of Arithmetic. Rules on Creation and Evaluation of Arithmetic and Logical Expressions. Scope of Data. Executable Statement Types. Assignment. Alphabetic Data Handling. Sequence Control and Decision Making Statements. Symbolic Data Handling. Inter-

action with Operating Systems and/or Equipment. Others. Declarations and Non-Executable Statements. Data Description. File Description. Format Description. Storage Allocation. Environment or Operating System Descriptions. Procedure, Subroutine, Function Declarations. Compiler Directives. Others. Structure of Language and Compiler Interaction. Self Modification of Programs. Self-Extension of the Language. Ability to Write the Compiler for a Language in that Language. Effect of Language Design on Implementation Efficiency. Debugging Aids and Error Checking. Other Features Not Included Above.

IV. LANGUAGES FOR NUMERICAL SCIENTIFIC PROBLEMS.

Scope of Chapter. Languages of Historical Interest Only. Very Early Systems. More Widely Used Systems. FORTRAN. History of FORTRAN. Functional Characteristics of ASA (USASI) FORTRAN and Basic FORTRAN. Technical Characteristics of ASA (USASI) Basic FORTRAN. Technical Features of ASA (USASI) FORTRAN. Significant Contribution to Technology. Significant Extensions of FORTRAN. ALGOL. History of ALGOL. Functional Characteristics of Revised ALGOL 60-Proposed ISO Standard. Technical Characteristics of Revised ALGOL 60-Proposed ISO Standard. Significant Contribution to Technology. Extensions of ALGOL. Languages Motivated by ALGOL 58. NELIAC. MAD. JOVIAL (cross reference only). On Line Languages. Introductory Remarks. JOSS (trademarked). QUIKTRAN. BASIC. CPS. MAP. Lincoln Reckoner. APL/360 and PAT. Culler-Fried. DIALOG. AMTRAN. Languages with Fairly Natural Mathematical Notation. Introductory Remarks. COLASL. MADCAP. MIRFAC. Klerer-May. Miscellaneous. CORC. OMNITAB. "More Non-Procedural."

V. LANGUAGES FOR BUSINESS DATA PROCESSING PROBLEMS.

Scope of Chapter. Languages of Primarily Historical Interest. FLOW-MATIC (and B-φ). AIMACO Commercial Translator. FACT. GECOM. COBOL. History of COBOL. Functional Characteristics of COBOL. Technical Characteristics of COBOL. Significant Contribution to Technology. File Handling. Extensions of COBOL. General (cross reference only).

VI. STRING AND LIST PROCESSING LANGUAGES.

Scope of Chapter. Languages of Historical Interest Only. IPL-V. History of IPL-V. Functional Characteristics of IPL-V. Technical Characteristics of IPL-V. Significant Contribution to Technology. L⁶. LISP 1.5. History of LISP 1.5. Functional Characteristics of LISP 1.5. Technical Characteristics of LISP 1.5. Significant Contribution to Technology. COMIT. History of COMIT. Functional Characteristics of COMIT. Technical Characteristics of COMIT. Significant Contribution to Technology. SNOBOL. History of SNOBOL.

Functional Characteristics of SNOBOL. Technical Characteristics of SNOBOL. Significant Contribution to Technology. TRAC (trademarked). Languages Not Widely Used. AMBIT. TREET. Others.

VII. FORMAL ALGEBRAIC MANIPULATION LANGUAGES.

Scope of Chapter. Languages of Historical Interest Only. ALGY. FORMAC. History of FORMAC. Functional Characteristics of FORMAC. Technical Features of FORMAC. Significant Contribution to Technology. MATHLAB. History of MATHLAB. Functional Characteristics of MATHLAB. Technical Characteristics of MATHLAB. Significant Contribution to Technology. ALTRAN. FLAP. Systems Requiring Special Equipment. Magic Paper. Symbolic Mathematical Laboratory.

VIII. MULTI-PURPOSE LANGUAGES.

Scope of Chapter. Languages of Historical Interest Only. JOVIAL. History of JOVIAL. Functional Characteristics of JOVIAL. Technical Characteristics of JOVIAL. Significant Contribution to Technology. PL/I. History of PL/I. Functional Characteristics of PL/I. Technical Characteristics of PL/I. Significant Contribution to Technology. Formula ALGOL. LISP 2.

IX. SPECIALIZED LANGUAGES.

Scope of Chapter. Languages for Special Application Areas. Machine Tool Control. Civil Engineering. Logical Design. Digital Simulation of Block Diagrams. Compiler Writing. Miscellaneous. Specialized Languages Across Application Areas. Discrete Simulation. Query. Graphics. Computer Aided Design. Text Editing. Control Languages for On-Line and Operating Systems.

X. SIGNIFICANT UNIMPLEMENTED CONCEPTS.

Scope of Chapter. UNCOL. Information Algebra. APL (Iverson). English. Hardware Implementation of Higher Level Languages.

XI. FUTURE LONG-RANGE DEVELOPMENTS.

Introduction. Theory Oriented Category. Language Definition, Translation and Creation. Next Major Conceptual Step. Non-Procedural Languages. Problem Describing Languages. Use of Mathematical Concepts. User Oriented Category. User Defined Language. Use of Natural Language. Communication with Hardware and Software. Languages for New Application Areas. Languages for Writing Software. Interrelationships Among Some of These Concepts. Conclusions and Summary.

Appendix A. BIBLIOGRAPHY ARRANGEMENTS AND AUTHOR LIST. Bibliography. General. Related Information. Author List.

Appendix B. LANGUAGE SUMMARY (list of languages with acronym, section in which it is discussed, very brief description, and one or two best references).

There is also a general bibliography embracing general items and most frequently cited items. And a bibliography of related information.

The table of contents is actually a superbly detailed outline of the structure of the programming language field.

IT'S READER ORIENTED AND MAKES YOUR LANGUAGE MORE MEANINGFUL

In the field of programming, where there is a plethora of languages, a certain amount of confusion and miscomprehension may well occur. You know this, and have probably experienced it. But PROGRAMMING LANGUAGES: History and Fundamentals alleviates the complexity of the situation for you by separating each language from the mass, and allowing you to examine and learn about it in isolation.

There are also sample programs for about 30 languages, showing you their basic elements at a glance.

AN INDISPENSABLE TOOL (AND AID) FOR YOU

You can readily discern the tremendous scope and utility the book possesses. A scope the magnitude of which only someone deeply involved in the environment of programming could foresee and prepare. And indeed the author is.

Jean Sammet, Programming Technology Manager (Federal Systems Division) IBM Corporation, brings to her multitude of papers, lectures, teaching assignments and this book, breadth and insight into programming languages. Plus an effectiveness in presenting the languages and making them more meaningful and comprehensible for you.

Study the table of contents on pages 3 & 4. It will offer you proof of the scope and value of this book. Then complete the enclosed order card and send for your free 15 day examination copy of this encyclopedic work.

PRE-PUBLICATION COMMENTS

from experts who examined the description of their specific language.

Concerning IPL, Allen Newell says, "The description seems accurate and complete enough for the level of treatment you [the author] are aiming at."

Dr. Alan J. Perlis says of the Formula ALGOL treatment, "I think you [the author] have captured the general intent of the design, its implementation, and its use in our university environment."

Chris Shaw says, "I thought it [JOVIAL] was a competent, objective piece of work."

And Melvin Klerer views the entire book as "... a unique compendium, highly recommended for all those interested in the design and comparative evaluation of programming languages."

April 1969

approx. 816 pp.

illus.

6 x 9"

Order from

PRENTICE HALL, Englewood Cliffs, N.J. 07632

PROGRAMMING LANGUAGES: History and Fundamentals

by Jean E. Sammet

A unique, definitive sourcebook presenting you with 120 programming languages! It's PROLIFIC, including and discussing all of the major, and most of the minor ones developed in the U.S.

It's UNIQUE, bringing together in one volume, fundamental data on programming languages enabling you to understand and write simple programs.

It's REFERENCE-ORIENTED, containing an extensive (over 800 items) bibliography grouped by language.

← See over for Table of Contents

NEW model-building and programming techniques for solving all types of simulation problems!

SYSTEM SIMULATION

by Geoffrey Gordon

Please send me a free 15-day examination copy of this vital, comprehensive sourcebook. After I have discovered for myself how it will strengthen my ability to meet and solve every type of simulation problem, and provide me with a full, firm introduction into the basics of simulation languages, techniques, applications, I'll remit \$11.50 plus postage and handling. Otherwise, after 15 days, I'll return the book and owe nothing.

I would also like a free 15 days examination copy of:

PROGRAMMING LANGUAGES, Pub. Apr. '69, 816 pp., illus., 6 × 9" (72998-8) \$18.00

Name _____

Address _____

City _____ State _____ Zip _____

SAVE! If payment accompanies order, publisher pays postage and handling charges
Same return privilege, refund guaranteed.

Dept. 1

28180-5

D-4887-LY(6)

Postage
Will be Paid
by
Addressee

No
Postage Stamp
Necessary
If Mailed in the
United States

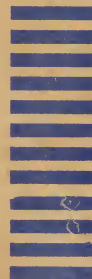
BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 365, ENGLEWOOD CLIFFS, N. J.

PRENTICE-HALL, INC.

ENGLEWOOD CLIFFS, NEW JERSEY

07632



**PROGRAMMING LANGUAGES:
History and Fundamentals**

- 1—General Introduction
 - 2—Functional Characteristics of Programming Languages
 - 3—Technical Characteristics of Programming Languages
 - 4—Languages for Numerical Scientific Problems
 - 5—Languages for Business Data Processing Problems
 - 6—String and List Processing Languages
 - 7—Formal Algebraic Manipulation Languages
 - 8—Multi-Purpose Languages
 - 9—Specialized Languages
 - 10—Significant Unimplemented Concepts
 - 11—Future Long-Range Developments
- Appendices